A Variance Reduced Stochastic Newton Method

Aurelien Lucchi **Brian McWilliams** **Thomas Hofmann**
Department of Computer Science, ETH Zürich
{aurelien.lucchi, brian.mcwilliams, thomas.hofmann } @inf.ethz.ch

Abstract

Quasi-Newton methods are widely used in practise for convex loss minimization problems. These methods exhibit good empirical performance on a wide variety of tasks and enjoy super-linear convergence to the optimal solution. For large-scale learning problems, stochastic Quasi-Newton methods have been recently proposed. However, these typically only achieve sub-linear convergence rates and have not been shown to consistently perform well in practice since noisy Hessian approximations can exacerbate the effect of high-variance stochastic gradient estimates. In this work we propose VITE, a novel stochastic Quasi-Newton algorithm that uses an existing first-order technique to reduce this variance. Without exploiting the specific form of the approximate Hessian, we show that VITE reaches the optimum at a geometric rate with a constant step-size when dealing with smooth strongly convex functions. Empirically, we demonstrate improvements over existing stochastic Quasi-Newton and variance reduced stochastic gradient methods.

1 Introduction

We consider the problem of optimizing a function expressed as an expectation over a set of data-dependent functions. Stochastic gradient descent (SGD) has become the method of choice for such tasks as it only requires computing stochastic gradients over a small subset of datapoints [2, 18]. The simplicity of SGD is both its greatest strength and weakness. Due to the effects of evaluating noisy approximation of the true gradient, SGD achieves a convergence rate which is only sub-linear in the number of steps. In an effort to deal with this randomness, two primary directions of focus have been developed. The first line of work focuses on choosing the appropriate SGD step-size [1, 10, 14]. If a decaying step-size is chosen, the variance is forced to zero asymptotically guaranteeing convergence. However, small steps also slow down progress and limit the rate of convergence in practise. The step-size must be chosen carefully, which can require extensive experimentation possibly negating the computational speedup of SGD. Another approach is to use an improved, lower-variance estimate of the gradient. If this estimator is chosen correctly – such that its variance goes to zero asymptotically – convergence can be guaranteed with a *constant* learning rate. This scheme is used in [5, 16] where the improved estimate of the gradient combines stochastic gradients computed at the current stage with others used at an earlier stage. A similar approach proposed in [8, 9] combines stochastic gradients with gradients periodically re-computed at a pivot point.

With variance reduction, first-order methods can obtain a linear convergence rate. In contrast, second-order methods have been shown to obtain super-linear convergence. However, this requires the computation and inversion of the Hessian matrix which is impractical for large-scale datasets. Approximate variants known as quasi-Newton methods [6] have thus been developed, such as the popular BFGS or its limited memory version known as LBFGS [11]. Quasi-Newton methods such as BFGS do not require computing the Hessian matrix but instead construct a quadratic model of the objective function by successive measurements of the gradient. This also yields super-linear convergence when the quadratic model is accurate. Stochastic variants of BFGS have been proposed (oBFGS [17]), for which stochastic gradients replace their deterministic counterparts. A regularized version known as RES [12] achieves a sublinear convergence rate with a decreasing step-size by enforcing a bound on the eigenvalues of the approximate Hessian matrix. SQN [3], another related method also requires a decreasing step size to achieve sub-linear convergence. Although stochastic second order methods have not be shown to achieve super-linear convergence, they empirically outperform SGD for problems with a large condition number [12].

A clear drawback to stochastic second order methods is that similarly to their first-order counterparts, they suffer from high variance in the approximation of the gradient. Additionally, this problem can be exaggerated due to the estimate of the Hessian magnifying the effect of this noise. Overall, this can lead to such algorithms taking large steps in poor descent directions.

In this paper, we propose and analyze a stochastic variant of BFGS that uses a multi-stage scheme similar to [8, 9] to progressively reduce the variance of the stochastic gradients. We call this method Variance-reduced Stochastic Newton (VITE). Under standard conditions on $\hat{J}$, we show that that variance reduction on the gradient estimate alone is sufficient for fast convergence. For smooth and strongly convex functions, VITE reaches the optimum at a geometric rate with a constant step-size. To our knowledge VITE is the first stochastic Quasi-Newton method with these properties.

In the following section, we briefly review the BFGS algorithm and its stochastic variants. We then introduce the VITE algorithm and analyze its convergence properties. Finally, we present experimental results on real-world datasets demonstrating its superior performance over a range of competitors.

2 Stochastic second order optimization

2.1 Problem setting

Given a dataset $\mathcal{D} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ consisting of feature vectors $\mathbf{x}_i \in \mathbb{R}^d$ and targets $y_i \in [0, C]$, we consider the problem of minimizing the expected loss $f(\mathbf{w}) = \mathbb{E}[f_i(\mathbf{w})]$. Each function $f_i(\mathbf{w})$ takes the form $f_i(\mathbf{w}) = \ell(h(\mathbf{w}, \mathbf{x}_i), y_i)$, where ℓ is a loss function and h is a prediction model parametrized by $\mathbf{w} \in \mathbb{R}^d$. The expectation is over the set of samples and we denote $\mathbf{w}^* = \arg\min_{\mathbf{w}} f(\mathbf{w})$.

This optimization problem can be solved exactly for convex functions using gradient descent, where the gradient of the loss function is expressed as $\nabla_{\mathbf{w}} f(\mathbf{w}) = \mathbb{E}[\nabla_{\mathbf{w}} f_i(\mathbf{w})]$. When the size of the dataset n is large, the computation of the gradient is impractical and one has to resort to stochastic gradients. Similar to gradient descent, stochastic gradient descent updates the parameter vector $\mathbf{w}_t$ by stepping in the opposite direction of the stochastic gradient $\nabla_{\mathbf{w}} f_i(\mathbf{w}_t)$ by an amount specified by a step size η_t as follows:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \nabla_{\mathbf{w}} f_i(\mathbf{w}_t). \tag{1}$$

In general, a stochastic gradient can also be computed as an average over a sample of datapoints as $\hat{f}(\mathbf{w}_t) = r^{-1} \sum_{i=1}^{r} f_i(\mathbf{w}_t)$. Given that the stochastic gradients are unbiased estimates of the gradient, Robbins and Monro [15] proved convergence of SGD to $\mathbf{w}^*$ assuming a decreasing step-size sequence. A common choice for the step size is [18, 12]

$$a)\ \eta_t = \frac{\eta_0}{t} \qquad \text{or} \qquad b)\ \eta_t = \frac{\eta_0\, T_0}{T_0 + t} \tag{2}$$

where η_0 is a constant initial step size and T_0 controls the speed of decrease.

Although the cost per iteration of SGD is low, it suffers from slow convergence for certain ill-conditioned problems [12]. An alternative is to use a second order method such as Newton's method that estimates the curvature of the objective function and can achieve quadratic convergence. In the following, we review Newton's method and its approximations known as quasi-Newton methods.

2.2 Newton's method and BFGS

Newton's method is an iterative method that minimizes the Taylor expansion of $f(\mathbf{w})$ around $\mathbf{w}_t$:

$$f(\mathbf{w}) = f(\mathbf{w}_t) + (\mathbf{w} - \mathbf{w}_t)^\top \nabla_{\mathbf{w}} f(\mathbf{w}_t) + \frac{1}{2}(\mathbf{w} - \mathbf{w}_t)^\top H (\mathbf{w} - \mathbf{w}_t), \tag{3}$$

where H is the Hessian of the function $f(\mathbf{w})$ and quantifies its curvature. Minimizing Eq. 3 leads to the following update rule:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t H_t^{-1} \cdot \nabla f(\mathbf{w}_t), \tag{4}$$

where η_t is the step size chosen by backtracking line search.

Given that computing and inverting the Hessian matrix is an expensive operation, approximate variants of Newton's method have emerged, where H_t^{-1} is replaced by an approximate version $\tilde{H}_t^{-1}$ selected to be positive definite and as close to H_t^{-1} as possible. The most popular member of this class of quasi-Newton methods is BFGS [13] that incrementally updates an estimate of the inverse Hessian, denoted $J_t = \tilde{H}_t^{-1}$. This estimate is computed by solving a weighted Frobenius norm minimization subject to the secant condition:

$$\mathbf{w}_{t+1} - \mathbf{w}_t = J_{t+1}(\nabla f(\mathbf{w}_{t+1}) - \nabla f(\mathbf{w}_t)). \tag{5}$$

The solution can be obtained in closed form leading to the following explicit expression:

$$J_{t+1} = \left(I - \frac{sy^\top}{y^\top s}\right) J_t \left(I - \frac{ys^\top}{y^\top s}\right) + \frac{ss^\top}{y^\top s}, \tag{6}$$

where $s = \mathbf{w}_{t+1} - \mathbf{w}_t$ and $y = \nabla f(\mathbf{w}_{t+1}) - \nabla f(\mathbf{w}_t)$. Eq. 6 is known to be positive definitive assuming that J_0 is initialized to be a positive definite matrix.

2.3 Stochastic BFGS

A stochastic version of BFGS (oBFGS) was proposed in [17] in which stochastic gradients are used for both the determination of the descent direction and the approximation of the inverse Hessian. The oBFGS approach described in Algorithm 1 uses the following update equation:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \hat{J}_t \cdot \nabla \hat{f}(\mathbf{w}_t), \tag{7}$$

where the matrix $\hat{J}_t$ and the vector $\nabla \hat{f}(\mathbf{w}_t)$ are stochastic estimates computed as follows. Let $\mathcal{A} \subset \{1 \dots n\}$ and $\mathcal{B} \subset \{1 \dots n\}$ be sets containing two independent samples of datapoints. The variables y and $\nabla f(\mathbf{w})$ defined in Eq. 6 are replaced by sampled variables computed as

$$\hat{y} = \frac{1}{|\mathcal{A}|} \sum_{k \in \mathcal{A}} \nabla f_k(\mathbf{w}_{t+1}) - \nabla f_k(\mathbf{w}_t) \quad \text{and} \quad \nabla \hat{f}(\mathbf{w}_t) = \nabla f_{\mathcal{B}}(\mathbf{w}_t) = \frac{1}{|\mathcal{B}|} \sum_{k \in \mathcal{B}} \nabla f_k(\mathbf{w}_t). \tag{8}$$

The estimate of the inverse Hessian then becomes

$$\hat{J}_{t+1} = \left(I - \frac{s\hat{y}^\top}{\hat{y}^\top s}\right) \hat{J}_t \left(I - \frac{\hat{y}s^\top}{\hat{y}^\top s}\right) + \frac{ss^\top}{\hat{y}^\top s} \tag{9}$$

Unlike Newton's method, oBFGS uses a fixed step size sequence instead of a line search. A common choice is to use a step size similar to the one used for SGD in Eq. 2.

A regularized version of oBFGS (RES) was recently proposed in [12]. RES differs from oBFGS in the use of a regularizer to enforce a bound on the eigenvalues of $\hat{J}_t$ such that

$$\gamma I \preceq \hat{J}_t \preceq \rho I = \left(\gamma + \frac{1}{\delta}\right) I, \tag{10}$$

where γ and δ are given positive constants and the notation $A \preceq B$ means that $B - A$ is a positive semi-definite matrix. Note that (10) also implies an upper and lower bound on $\mathbb{E}[\hat{J}_t]$ [12]. The update of RES is modified to incorporate an identity bias term γI as follows:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t(\hat{J}_t + \gamma I) \cdot \nabla \hat{f}(\mathbf{w}_t). \tag{11}$$

The convergence proof derived in [12] shows that lower and upper bounds on the Hessian eigenvalues of the sample functions are sufficient to guarantee convergence to the optimum. However, the analysis shows that RES will converge to the optimum at a rate $\mathcal{O}(1/t)$ and requires a decreasing step-size. Similar results were derived in [3] for the SQN algorithm.

Algorithm 1 oBFGS

```
INPUTS :
    $\mathcal{D}$ : Training set of $n$ examples.
    $\mathbf{w}_0$ : Arbitrary initial values, e.g., 0.
    $\{\eta_t\}$ : Step size sequence
OUTPUT : $\mathbf{w}_t$
$\hat{J}_0 \leftarrow \alpha I$
for $t = 0 \dots T$ do
    Randomly pick two sets $\mathcal{A}$ and $\mathcal{B}$
    $s \leftarrow \mathbf{w}_{t+1} - \mathbf{w}_t$
    $\hat{y} \leftarrow \sum_{k \in \mathcal{B}} \nabla f_k(\mathbf{w}_{t+1}) - \nabla f_k(\mathbf{w}_t)$
    $\nabla \hat{f}(\mathbf{w}_t) \leftarrow \sum_{k \in \mathcal{A}} \nabla f_k(\mathbf{w}_t)$
    $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta_t \hat{J}_{t+1} \cdot \nabla \hat{f}(\mathbf{w}_t)$
    $\hat{J}_{t+1} \leftarrow \left(I - \frac{s\hat{y}^\top}{\hat{y}^\top s}\right) \hat{J}_t \left(I - \frac{\hat{y}s^\top}{\hat{y}^\top s}\right) + \frac{ss^\top}{\hat{y}^\top s}$
end for
```

3 The VITE algorithm

Reducing the size of the sets $\mathcal{A}$ and $\mathcal{B}$ used to estimate the inverse Hessian approximation and the stochastic gradient is desirable for reasons of computational efficiency. However, doing so also increases the variance of the update step. Here we propose a new method called VITE that explicitly reduces this variance.

In order to simplify the analysis of VITE, we do not explicitly consider the randomness in the matrix $\hat{J}_t$. Instead, we assume that it is positive definite (which holds under weak conditions due to the BFGS update step) and that its variance can be kept under control, for example by using the regularization of the RES method.

To motivate VITE we first consider the standard oLBFGS step, (7) estimated with the sets $\mathcal{A}$ and $\mathcal{B}$. The first and second moments simplify as

$$\mathbb{E}\,[\hat{J}_t \nabla f_{\mathcal{B}}(\mathbf{w}_t)] = \hat{J}_t \mathbb{E}_{\mathcal{B}}[\nabla f_{\mathcal{B}}(\mathbf{w}_t)] \tag{12}$$

and

$$\mathbb{E}\,\left|\left|\hat{J}_t \nabla f_{\mathcal{B}}(\mathbf{w}_t)\right|\right|^2 \leq \left|\left|\hat{J}_t\right|\right|^2 \mathbb{E}_{\mathcal{B}}\,||\nabla f_{\mathcal{B}}(\mathbf{w}_t)||^2\,, \tag{13}$$

respectively. For $|\mathcal{A}|$ large enough, in order to reduce the variance of the estimate $\hat{J}_t \cdot \nabla f_{\mathcal{B}}(\mathbf{w}_t)$, it is only required to reduce the variance of $\nabla f_{\mathcal{B}}(\mathbf{w}_t)$ independently. We proceed using a technique similar to the one proposed in [8, 9].

VITE differs from oBFGS and other stochastic Quasi-Newton methods in the use of a multi-stage scheme as shown in Algorithm 2. In the outer loop a variable $\tilde{\mathbf{w}}$ is introduced. We periodically evaluate the gradient of the function with respect to $\tilde{\mathbf{w}}$. This *pivot point* is inserted in the update equation to reduce the variance. Each inner loop runs for a a random number of steps $t_j \in [1, m]$ whose distribution follows a geometric law with parameter $\beta = \sum_{t=1}^{m}(1 - \mu\gamma\eta)^{m-t}$. Stochastic gradients at $\mathbf{w}_t$ and $\tilde{\mathbf{w}}$ are computed and the inverse Hessian approximation is updated in each iteration of the inner loop. $\hat{J}_t$ can be updated using the same update as RES although we found in practice that using Eq. 9 did not affect the results significantly. The descent direction $\nabla f_{\mathcal{B}}(\mathbf{w})$ is then replaced by

$$\mathbf{v}_t = \nabla f_{\mathcal{B}}(\mathbf{w}_t) - \nabla f_{\mathcal{B}}(\tilde{\mathbf{w}}) + \tilde{\nu}.$$

VITE then makes updates of the form

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \hat{J}_t \cdot \mathbf{v}_t. \tag{14}$$

Clearly, $\tilde{\nu} = \mathbb{E}[\nabla f_{\mathcal{B}}(\tilde{\mathbf{w}})]$ and $\mathbb{E}[\mathbf{v}_t] = \mathbb{E}[\nabla f_{\mathcal{B}}(\mathbf{w}_t)]$ so in expectation the descent is in the same direction as Eq. (12). Following the analysis of [8], the variance of $\mathbf{v}_t$ goes to zero when both $\tilde{\mathbf{w}}$ and $\mathbf{w}_t$ converge to the same parameter $\mathbf{w}^*$. Therefore, convergence can be guaranteed with a *constant* step-size. The complexity of this approach depends on the number of epochs S and a constant m limiting the number of stochastic gradients computed in a single epoch, as well as other parameters that will be introduced in more detail in Section 4.

Algorithm 2 VITE

```
INPUTS :
  $\mathcal{D}$ : Training set of $n$ examples        $\tilde{\mathbf{w}}_0$ : Arbitrary initial values, e.g., 0
  $\eta$ : Constant step size                       $m$: Arbitrary constant
OUTPUT : $\mathbf{w}_t$
$\hat{J}_0 \leftarrow \alpha I$
for $s = 0 \dots S$ do
  $\tilde{\mathbf{w}} = \tilde{\mathbf{w}}_{s-1}$
  $\tilde{\nu} = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\tilde{\mathbf{w}})$
  $\mathbf{w}_0 = \tilde{\mathbf{w}}$
  Let $t_j \leftarrow t$ with probability $\frac{(1-\mu\rho\eta)^{m-t}}{\beta}$ for $t = 1, \dots, m$
  for $t = 0 \dots t_j - 1$ do
    Randomly pick independent sets $\mathcal{A}, \mathcal{B} \subset \{1 \dots n\}$
    $\mathbf{v}_t = \nabla f_{\mathcal{B}}(\mathbf{w}_t) - \nabla f_{\mathcal{B}}(\tilde{\mathbf{w}}) + \tilde{\nu}$
    $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta \hat{J}_t \cdot \mathbf{v}_t$
    Update $\hat{J}_{t+1}$
  end for
  $\tilde{\mathbf{w}}_s = \mathbf{w}_{t_j}$.
end for
```

4 Analysis

In this section we present a convergence proof for the VITE algorithm that builds upon and generalizes previous analyses of variance reduced first order methods [8, 9]. Specifically, we show how variance reduction on the stochastic gradient direction is sufficient to establish geometric convergence rates, even when performing linear transformations with a matrix $\hat{J}_t$. Since we do not exploit the specific form of the stochastic evolution equations for $\hat{J}_t$, this analysis will not allow us to argue in favor of the specific choice of Eq. (9), yet it shows that variance reduction on the gradient estimate is sufficient for fast convergence as long as $\hat{J}_t$ is sufficiently well behaved. Our analysis relies on the following standard assumptions:

A1 Each function f_i is differentiable and has a Lipschitz continuous gradient with constant $L > 0$, i.e. $\forall \mathbf{w}, \mathbf{v} \in \mathbb{R}^n$,

$$f_i(\mathbf{w}) \leq f_i(\mathbf{v}) + (\mathbf{w} - \mathbf{v})^\top \nabla f_i(\mathbf{v}) + \frac{L}{2} ||\mathbf{w} - \mathbf{v}||^2 \tag{15}$$

A2 f is μ-strongly convex, i.e. $\forall \mathbf{w}, \mathbf{v} \in \mathbb{R}^n$,

$$f(\mathbf{w}) \geq f(\mathbf{v}) + (\mathbf{w} - \mathbf{v})^\top \nabla f(\mathbf{v}) + \frac{\mu}{2} ||\mathbf{w} - \mathbf{v}||^2 \tag{16}$$

which also implies

$$||\nabla f(\mathbf{w})||^2 \geq 2\mu(f(\mathbf{w}) - f(\mathbf{w}^*)) \ \ \forall \mathbf{w} \in \mathbb{R}^n \tag{17}$$

for the minimizer $\mathbf{w}^*$ of f.

Assumptions **A1** and **A2** also implies that the eigenvalues of the Hessian are bounded as follows:

$$\mu I \preceq H_t \preceq LI. \tag{18}$$

Finally we make the assumption that the inverse Hessian approximation is always well-behaved.

A3 There exist positive constants γ and ρ such that, $\forall \mathbf{w} \in \mathbb{R}^n$,

$$\gamma I \preceq \hat{J}_t \preceq \rho I. \tag{19}$$

Assumption **A3** is equivalent to assuming that $\hat{J}_t$ is bounded in expectation (see: e.g. [12]) but allows us to remove this complication, simplifying notation in the analysis which follows. We now introduce two lemmas required for the proof of convergence.

Lemma 1. *The following identity holds:*

$$\mathbb{E}f(\tilde{\mathbf{w}}_{s+1}) = \frac{1}{\beta}\sum_{t=0}^{m-1}\tau_t \mathbb{E}f(\mathbf{w}_t)$$

where $\tau_t := (1-\gamma\eta\mu)^{m-t-1}$ and the weight vectors $\mathbf{w}_t$ belong to epoch s.

This result follows directly from Lemma 3 in [9].

Lemma 2.

$$\mathbb{E}\|\mathbf{v}_t\|^2 \leq 4L(f(\mathbf{w}_t) - f(\mathbf{w}^*) + f(\tilde{\mathbf{w}}) - f(\mathbf{w}^*))$$

The proof is given in [8, 9] and reproduced for convenience in the Appendix. We are now ready to state our main result.

Theorem 1. *Let Assumptions* **A1**-**A3** *be satisfied. Define the rescaled strong convexity $\mu' := \gamma\mu \leq \mu$ and Lipschitz $L' := \rho L \geq L$ constants respectively. Choose $0 < \eta \leq \frac{\mu'}{2L'^2}$ and let m be sufficiently large so that $\alpha = \frac{(1-\eta\mu')^m}{\beta\eta(\mu'-2L'^2\eta)} + \frac{2L'^2\eta}{\mu'-2L'^2\eta} < 1$.*

Then the suboptimality of $\tilde{\mathbf{w}}_s$ is bounded in expectation as follows:

$$\mathbb{E}(f(\tilde{\mathbf{w}}_s) - f(\mathbf{w}^*) \leq \alpha^s \mathbb{E}[f(\mathbf{w}_0) - f(\mathbf{w}^*)]. \tag{20}$$

Remark 1. *Observe that γ and ρ are bounds on the* inverse *Hessian approximation. If $\hat{J}_t$ is a good approximation to H, then by plugging in $\gamma = L$ and $\rho = \mu$, the upper bound on the learning rate reduces to $\eta \leq \frac{1}{2\mu L}$.*

Proof of Theorem 1. Our starting point is the basic inequality

$$\begin{aligned} f(\mathbf{w}_{t+1}) &= f(\mathbf{w}_t - \eta\hat{J}_t \cdot \mathbf{v}_t) \\ &\leq f(\mathbf{w}_t) - \eta\langle\nabla f(\mathbf{w}_t), \hat{J}_t \cdot \mathbf{v}_t\rangle + \frac{L}{2}\eta^2\left|\left|\hat{J}_t\mathbf{v}_t\right|\right|^2. \end{aligned} \tag{21}$$

We first use the properties of $\mathbf{v}_t$ and $\hat{J}_t$ to reduce the dependence of (21) on $\hat{J}_t$ to its largest and smallest eigenvalues given by (19). For the purpose of the analysis, we define $\mathcal{F}_t$ to be the sigma-algebra measuring $\mathbf{w}_t$. By conditioning on $\mathcal{F}_t$, and by **A3**, the remaining randomness is in the choice of the index set $\mathcal{B}$ in round t, which is tied to the stochasticity of $\mathbf{v}_t$. Taking expectations with respect to $\mathcal{B}$ gives us

$$\mathbb{E}_{\mathcal{B}}\left|\left|\hat{J}_t\mathbf{v}_t\right|\right|^2 \leq \|\hat{J}_t\|^2\mathbb{E}_{\mathcal{B}}\|\mathbf{v}_t\|^2 \leq \rho^2\mathbb{E}_{\mathcal{B}}\|\mathbf{v}_t\|^2 \tag{22}$$

and

$$\mathbb{E}_{\mathcal{B}}\langle\nabla f(\mathbf{w}_t), \hat{J}_t \cdot \mathbf{v}_t\rangle = \langle\nabla f(\mathbf{w}_t), \hat{J}_t \cdot \nabla f(\mathbf{w}_t)\rangle \geq \gamma\left|\left|\nabla f(\mathbf{w}_t)\right|\right|^2 \tag{23}$$

where (23) comes from the definition $\mathbb{E}_{\mathcal{B}}\mathbf{v}_t = \nabla f(\mathbf{w}_t)$. Therefore, taking the expectation of the inequality (21) and dropping the notational dependence on $\mathcal{B}$ results in

$$\mathbb{E}f(\mathbf{w}_{t+1}) \leq \mathbb{E}f(\mathbf{w}_t) - \gamma\eta\mathbb{E}\left|\left|\nabla f(\mathbf{w}_t)\right|\right|^2 + \frac{L}{2}\eta^2\rho^2\mathbb{E}\left|\left|\mathbf{v}_t\right|\right|^2. \tag{24}$$

To simplify the remainder of the proof we make the following substitution

$$\mu' := \gamma\mu \leq \mu \quad \text{and} \quad L' := \rho L \geq L.$$

Considering a fixed epoch s, we can further bound $\mathbb{E}f(\mathbf{w}_{t+1})$ using Lemma 2 and Eq. 17. By taking the expectation over $\mathcal{F}_t$, adding and subtracting $f(\mathbf{w}^*)$, we get

$$\begin{aligned} \mathbb{E}[f(\mathbf{w}_{t+1}) - f(\mathbf{w}^*)] \leq &\mathbb{E}[f(\mathbf{w}_t) - f(\mathbf{w}^*)] + 2\eta^2 L'^2\big(f(\tilde{\mathbf{w}}_s) - f(\mathbf{w}^*)\big) \qquad (25) \\ &+ 2\big(\eta^2 L'^2 - \eta\mu'\big)\mathbb{E}[f(\mathbf{w}_t) - f(\mathbf{w}^*)] \\ = &2\eta^2 L'^2\big(f(\tilde{\mathbf{w}}_s) - f(\mathbf{w}^*)\big) + \big(2\eta^2 L'^2 - 2\eta\mu' + 1\big)\mathbb{E}[f(\mathbf{w}_t) - f(\mathbf{w}^*)]. \end{aligned}$$

Writing $\Delta f(\mathbf{w}_t) := f(\mathbf{w}_t) - f(\mathbf{w}^*)$, we then have

$$(\eta\mu' - 2\eta^2 {L'}^2)\mathbb{E}\Delta f(\mathbf{w}_t) \leq 2\eta^2 {L'}^2 \Delta f(\tilde{\mathbf{w}}_s) + (1-\eta\mu')\mathbb{E}\Delta f(\mathbf{w}_t) - \mathbb{E}\Delta f(\mathbf{w}_{t+1}) \tag{26}$$

Now we sum all these inequalities at iterations $t = 0, \ldots, m-1$ performed in epoch s with weights $\tau_t = (1-\eta\mu')^{m-t-1}$. Applying Lemma 1 to the last summand to recover $f(\tilde{\mathbf{w}}_{s+1})$ we arrive at

$$\beta\mathbb{E}\Delta f(\tilde{\mathbf{w}}_{s+1}) \leq \frac{2\beta\eta^2 {L'}^2}{\eta\mu' - 2\eta^2 {L'}^2}\mathbb{E}\Delta f(\tilde{\mathbf{w}}_s) + \sum_{t=0}^{m-1} \tau_t \frac{(1-\eta\mu')\mathbb{E}\Delta f(\mathbf{w}_t) - \mathbb{E}\Delta f(\mathbf{w}_{t+1})}{\eta\mu' - 2\eta^2 {L'}^2}.$$

We now need to bound the remaining sum $(*)$ in the numerator, which can be accomplished by re-grouping summands

$$(*) = (1-\eta\mu')^m \mathbb{E}\triangle f(\tilde{\mathbf{w}}_s) - (1-\eta\mu')\mathbb{E}\triangle f(\tilde{\mathbf{w}}_{s+1})$$

By ignoring the negative term in $(*)$, we get the final bound

$$\mathbb{E}\Delta f(\tilde{\mathbf{w}}_{s+1}) \leq \alpha \mathbb{E}\Delta f(\tilde{\mathbf{w}}_s),$$

where

$$\alpha = \left(\frac{(1-\eta\mu')^m}{\beta(\eta\mu' - 2\eta^2 {L'}^2)} + \frac{2\eta^2 {L'}^2}{\eta\mu' - 2\eta^2 {L'}^2} \right)$$

□

Theorem 1 implies that VITE has a local geometric convergence rate with a constant learning rate. In order to satisfy $\mathbb{E}(f(\tilde{\mathbf{w}}_s) - f(\mathbf{w}^*)) \leq \epsilon$, the number of stages s needs to satisfy

$$s \geq -\log \alpha^{-1} \log \frac{\mathbb{E}(f(\tilde{\mathbf{w}}_0) - f(\mathbf{w}^*))}{\epsilon}.$$

Since each stage requires $n + m(2|\mathcal{A}| + 2|\mathcal{B}|)$ component gradient evaluations, the overall complexity is $\mathcal{O}((n + 2m(|\mathcal{A}| + |\mathcal{B}|))\log(1/\epsilon))$.

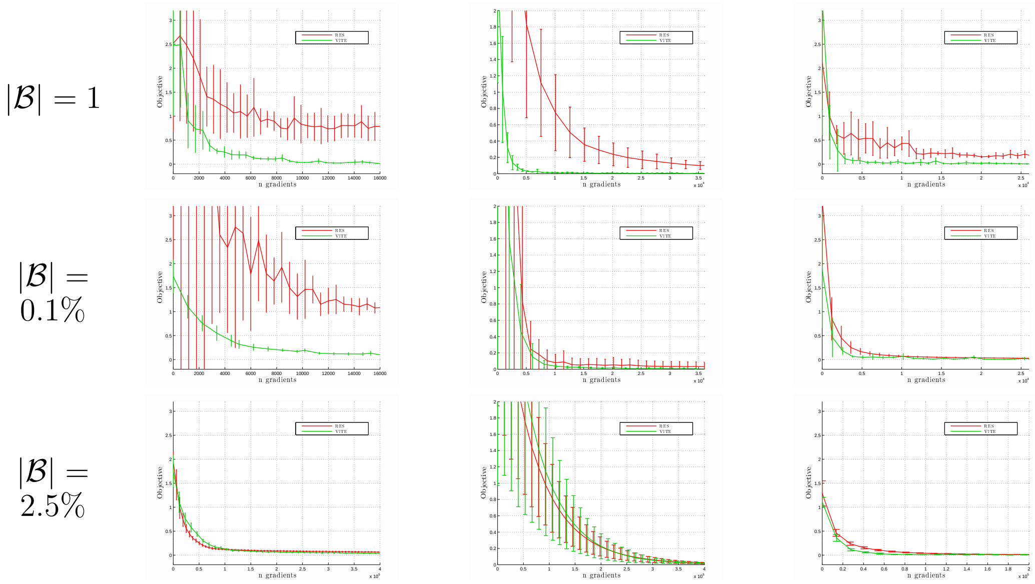
5 Experimental Results

This section presents experimental results that compare the performance of VITE to SGD, SVRG [8] which incorporates variance reduction and RES [12] which incorporates second order information. We consider two commonly occurring problems in machine learning, namely least-square regression and regularized logistic regression.

Linear Least Squares Regression. We apply least-square regression on the binary version of the COV dataset [4] that contains $n = 581,012$ datapoints, each described by $d = 54$ input features. **Logistic Regression.** We apply logistic regression on the ADULT and IJCNN1 datasets obtained from the LIBSVM website [1]. The ADULT dataset contains $n = 32,561$ datapoints, each described by $d = 123$ input features. The IJCNN1 dataset contains $n = 49,990$ datapoints, each described by $d = 22$ input features. We added an ℓ_2-regularizer with parameter $\lambda = 10^{-5}$ to ensure the objective is strongly convex.

The complexity of VITE depends on three quantities: the approximate Hessian $\hat{J}$, the pair of stochastic gradients $(\nabla f_{\mathcal{B}}(\mathbf{w}), \nabla f_{\mathcal{B}}(\tilde{\mathbf{w}}))$ and $\tilde{\nu}$, respectively computed over the sets $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{D}$. Similarly to [12], we consider different choices for $|\mathcal{A}|$ and $|\mathcal{B}|$ and pick the best value in a limited interval $\{1, \ldots, 0.05n\}$. These results are also reported for the RES method that also depends on both $|\mathcal{A}|$ and $|\mathcal{B}|$. For SGD, we use $|\mathcal{B}| = 1$ as we found this value to be the best performer on all datasets. Computing the average gradient, $\tilde{\nu}$ over the full dataset for SVRG and VITE is impractical. We therefore estimate $\tilde{\nu}$ over a small subset $\mathcal{C} \subset \mathcal{D}$. Although this introduces some bias, it did not seem to practically affect convergence for sufficiently large $|\mathcal{C}|$. In our experiments, we selected $|\mathcal{C}| = 0.1n$ samples uniformly at random. Each experiment was averaged over 5 runs with different

[1] `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets`



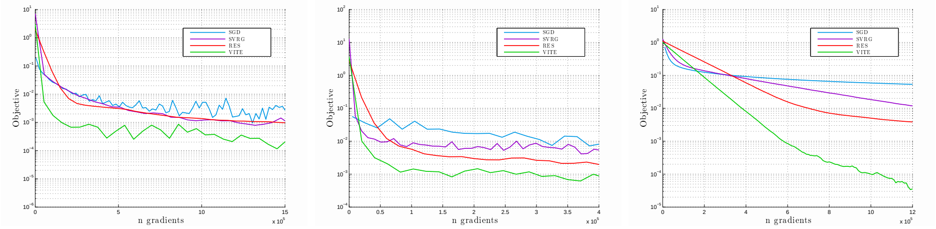
(a) COV (b) ADULT (c) IJCNN

Figure 1: *The red and green curves are the losses achieved by RES and* VITE *respectively for varying size of* $|\mathcal{B}|$ *as a percentage of* n*. Each experiment was averaged over 5 runs. Error bars denote variance. In the regime* $|\mathcal{B}| \leq 0.1\%$*,* VITE *has a much lower variance and reaches a lower optimum value. Increasing* $|\mathcal{B}|$ *further decreases the variance of the stochastic gradients but requires more gradient evaluations, decreasing the gap in performance between the methods. Overall, we found* VITE *with* $|\mathcal{B}| = 1\%$ *and* $|\mathcal{B}| = 0.1\%$ *to perform the best.*

initializations of $\mathbf{w}_0$ and a random selection of the samples in $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$. Given that the complexity per iteration of each method is different, we compare them as a function of the number of gradient evaluations.

Fig. 1 shows the empirical convergence properties of VITE against RES for least-square regression and logistic regression. The horizontal axis corresponds to the number of gradient evaluations while the vertical axis corresponds to the objective function value. The vertical bars in each plot show the variance over 5 runs. We show plots for different values of $|\mathcal{B}|$ and the best corresponding $\mathcal{A}$. For small $|\mathcal{B}|$, the variance of the stochastic gradients clearly hurts RES while the variance corrections of VITE lead to fast convergence. As we increase $|\mathcal{B}|$, thus reducing the variance of the stochastic gradients, the convergence rate of RES and VITE becomes similar. However, VITE with small $|\mathcal{B}|$ is much faster to converge to a lower objective value. This clearly demonstrates how using small batches for the computation of the gradients while reducing their variance leads to a fast convergence rate. We also investigated the effect of $|\mathcal{A}|$ on the convergence of RES and VITE (see Appendix). In short, we find that a good-enough curvature estimate can be obtained for $|\mathcal{A}| = \mathcal{O}(10^{-5}n)$. Increasing this value incurs a penalty in terms of number of gradient evaluations required and so overall performance degrades.

Finally, we compared VITE against SGD, RES and SVRG [8, 9]. A critical factor in the performance of SGD is the selection of the step-size. We use the step-size given in Eq. 2*b* and pick the parameters T_0 and η_0 by performing cross-validation over $T_0 = \{1, 10, 10^2, \ldots, 10^4\}$ and $\eta_0 = \{10^{-1}, \ldots, 10^{-5}\}$. Although it is a quasi-Newton method, RES also requires a decaying step-size and so the same selection process was performed. For SVRG and VITE, we used a *constant* step size chosen in the same interval as η_0. For SVRG and VITE we used the same size subset, $\mathcal{C}$ to compute $\tilde{\nu}$. Fig. 2 shows the objective value of each method in log scale. Although RES and SVRG are superior to SGD, neither clearly outperforms the other. On the other hand, we observe that VITE consistently converges faster than both RES and SVRG. This demonstrates that the combination of second order information *and* variance reduction is beneficial for fast convergence.



(a) COV (b) ADULT (c) IJCNN

Figure 2: *Comparison of RES and* VITE *(trained with the best performing parameters) against SGD and SVRG. The reduction in variance for* VITE *is faster than SGD or RES which typically lead to faster convergence.*

6 Conclusion

We have shown that stochastic variants of BFGS can be made more robust to the effects of noisy stochastic gradients using variance reduction. We introduced VITE and showed that it obtains a geometric convergence rate for smooth convex functions – to our knowledge the first stochastic Quasi-Newton algorithm with this property. We have shown experimentally that VITE outperforms both variance reduced SGD and stochastic BFGS. The theoretical analysis we present is quite general and additionally only requires that the bound on the eigenvalues of the inverse Hessian matrix in (19) holds. Therefore, the variance reduced framework we propose can be extended to other quasi-Newton methods, including the widely used L-BFGS and ADAGRAD [7] algorithms. Finally, an important open question is how to bridge the gap between the theoretical and empirical results. Specifically, whether it is possible to obtain better convergence rates for stochastic BFGS algorithms which match the improvement we have demonstrated over SVRG.

7 Appendix

7.1 Proof of Lemma 2

$$\begin{aligned}
&\mathbb{E}\,||\mathbf{v}_t||^2 = \mathbb{E}\,||\nabla f_i(\mathbf{w}_t) - \nabla f_i(\tilde{\mathbf{w}}) + \nabla f(\tilde{\mathbf{w}})||^2 \\
&\quad \leq 2\mathbb{E}\,||\nabla f_i(\mathbf{w}_t) - \nabla f_i(\mathbf{w}^*)||^2 \\
&\quad + 2\mathbb{E}\,||(\nabla f_i(\tilde{\mathbf{w}}) - \nabla f_i(\mathbf{w}^*)) - \nabla f(\tilde{\mathbf{w}})||^2 \\
&\quad = 2\mathbb{E}\,||\nabla f_i(\mathbf{w}_t) - \nabla f_i(\mathbf{w}^*)||^2 \\
&\quad + 2\mathbb{E}\,||(\nabla f_i(\tilde{\mathbf{w}}) - \nabla f_i(\mathbf{w}^*)) - (\nabla f(\tilde{\mathbf{w}}) - \nabla f(\mathbf{w}^*))||^2 \\
&\quad \leq 2\mathbb{E}\,||\nabla f_i(\mathbf{w}_t) - \nabla f_i(\mathbf{w}^*)||^2 \\
&\quad + 2\mathbb{E}\,||\nabla f_i(\tilde{\mathbf{w}}) - \nabla f_i(\mathbf{w}^*)||^2 \\
&\quad \leq 4L(f(\mathbf{w}_t) - f(\mathbf{w}^*) + f(\tilde{\mathbf{w}}) - f(\mathbf{w}^*))
\end{aligned} \tag{27}$$

The second inequality uses $\mathbb{E}\,||\xi - \mathbb{E}\xi||^2 = \mathbb{E}\,||\xi||^2 - ||\mathbb{E}\xi||^2 \leq \mathbb{E}\,||\xi||^2$ for any random vector ξ.

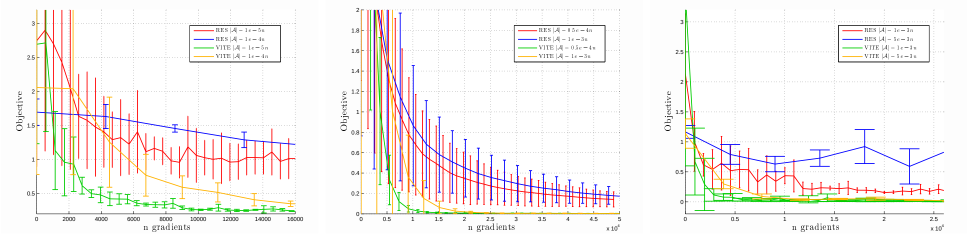
The last inequality uses the following inequality derived from the fact that f_i is a Lipschitz function:

$$\mathbb{E}\,||\nabla f_i(\mathbf{w}^*) - \nabla f_i(\mathbf{w}_t)||^2 \leq 2L(f(\mathbf{w}_t) - f(\mathbf{w}^*)).$$

□

7.2 Selection of the parameter $|\mathcal{A}|$.

Figure 3 shows the effect of the set $\mathcal{A}$, used to estimate the inverse Hessian, on the convergence of RES and VITE. We show results for $|\mathcal{A}| = \{0.00001,\ 0.0001\} \times n$. Firstly we see that better performance is obtained for both methods for the smaller value of $|\mathcal{A}|$. By increasing $|\mathcal{A}|$, the penalty paid in terms of gradient evaluations outweighs the gain in terms of better curvature estimates and so convergence is slower. A similar observation was made in [12]. However, we also observe that VITE always outperforms RES for all combinations of $|\mathcal{A}|$.



(a) COV (b) ADULT (c) IJCNN

Figure 3: Evolution of the objective value of RES and VITE for different values of $|\mathcal{A}|$. We can see that the lowest value of $|\mathcal{A}|$ performs better, which indicates than there is no gain at increasing this value passed a certain cut-off value.

References

[1] F. Bach, E. Moulines, et al. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Advances in Neural Information Processing Systems*, pages 451–459, 2011.

[2] L. Bottou. Large-scale machine learning with stochastic gradient descent. In *COMPSTAT*, pages 177–186. Springer, 2010.

[3] R. H. Byrd, S. Hansen, J. Nocedal, and Y. Singer. A stochastic quasi-newton method for large-scale optimization. *arXiv preprint arXiv:1401.7020*, 2014.

[4] R. Collobert, S. Bengio, and Y. Bengio. A parallel mixture of svms for very large scale problems. *Neural computation*, 14(5):1105–1114, 2002.

[5] A. Defazio, F. Bach, and S. Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pages 1646–1654, 2014.

[6] J. E. Dennis, Jr and J. J. Moré. Quasi-newton methods, motivation and theory. *SIAM review*, 19(1):46–89, 1977.

[7] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

[8] R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pages 315–323, 2013.

[9] J. Konečnỳ and P. Richtárik. Semi-stochastic gradient descent methods. *arXiv preprint arXiv:1312.1666*, 2013.

[10] S. Lacoste-Julien, M. Schmidt, and F. Bach. A simpler approach to obtaining an o (1/t) convergence rate for the projected stochastic subgradient method. *arXiv preprint arXiv:1212.2002*, 2012.

[11] D. C. Liu and J. Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

[12] A. Mokhtari and A. Ribeiro. Res: Regularized stochastic bfgs algorithm. *arXiv preprint arXiv:1401.7625*, 2014.

[13] J. Nocedal and S. Wright. *Numerical optimization*, volume 2. Springer New York, 1999.

[14] A. Rakhlin, O. Shamir, and K. Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. *arXiv preprint arXiv:1109.5647*, 2011.

[15] H. Robbins and S. Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[16] N. L. Roux, M. Schmidt, and F. R. Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. In *Advances in Neural Information Processing Systems*, pages 2663–2671, 2012.

[17] N. Schraudolph, J. Yu, and S. Günter. A stochastic quasi-newton method for online convex optimization. In *Intl. Conf. Artificial Intelligence and Statistics (AIstats)*, 2007.

[18] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter. Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical programming*, 127(1):3–30, 2011.